

Appl. No. 09 / 973,537  
Comm. Dated August 1st, 2006  
Reply To Office action of June 8th, 2006

## Remarks / Arguments

### Report of the amendments to the claims

-I have made the suggested corrections relating to formal matters in claims 8, 10, 15, 20, 22, 24 and 26.  
-I have amended the claim 9 according to suggested correction.  
-I have amended the claims 14, 16, 18, 19, 20 and 24 (the claim 24 was apparently forgotten from the list of rejected claims) to define the "other property" with simplest prior art terms. I have also provided justifying reasons for using any data variable of a packet as the process input. I have also provided evidence that the term property is recognized by the prior art, and that it suits well for describing data items in the data structure of data packets, and that the prior art knows processing properties of an object, like data packet properties.

### Patentability arguments

#### Arguments for the claims 14, 16, 18, 19, 20 and 24

The claims rely on the universality of the object model of modern object oriented programming and its direct applicability to data packets, complying with prior art process description conventions

In my view the computing technology prior art is specific enough in defining a property, and cultivated enough to understand what it is, and knowledgeable enough to follow instructions which recite its use. Anyway, it is not my own term, but a term defined and recognized by the prior art. The term property has been originally wider touted by Microsoft Corp. in its object oriented programming languages as a part of the object model, where the property is used to denote individual data items or attributes which have been established beforehand in the data structure of a class / object. I have included the term in the invention specification because data packets are simple good examples of independent data objects, the data structure of data packets bears much resemblance in technical specification and programming code to the data structure of class objects, the different data items and data fields of a data packet named and listed in a type declaration statement for the packet. Object properties are used routinely in every modern programming language. The Webopedia.com computing dictionary defines a property as "*Characteristic of an object. In many programming languages, including Visual Basic, the term property is used to describe attributes associated with a data structure.*". Visual Basic is one of Microsoft's programming languages.

Still I have voluntarily revised the claims with an elaborate definition for "other property", which hears now: "any data variable which has been specially assigned to guide the switching by its data value". The delivery address of a packet is a property of the packet object by prior art conventions, and it is also a data variable, what analogy is used in the definition for the "other property". I hope I am allowed to rely on prior art knowing what a data variable is, because it relates to age-old standard programming. Prior art organizes the data structure of a packet in data fields, of which most contain a data variable, and one contains the packet carried data. There should be no ambiguity about what the "any data variable" is, it is a data variable in the packet which is specially assigned to guide the switching (or routing in claim 20) by its data value. Note that the data variable is specially assigned, i.e. assigned beforehand, so the

processing, as subsumed by this terminology, uses a data variable which use and existence in the packet is predetermined, and which place in the packet is expressly specified, so that the data variable is individually identified and differentiated from other data and data variables in the packet. The use of a named data variable works the same way, it establishes a reference to the use, existence and specific place of a predetermined data field and its data variable in the packet according to prior art conventions. So my definition for the "other property" is even overly loyal to the prior art conventions, using the simplest basic terms in its definition in a correct prior art recognized way.

If there is a problem with justifying the use of "any" data variable without specifically naming it, I would like to note that the prior art for example allows the database routine specifications to use "any data field" expression for a description of a database record sorting algorithm in e.g. referring to the sort key field, without having to name the specific purposes for which the possible sort key fields of a record are established (like storing an employee's name, address, gender, age). Similarly a data packet contains informational data variables organized in data fields, and the variables can be treated uniformly in the packet processing procedures described in my claims, regardless of what specific data variable is selected as an input of the procedure; this should be no complicated issue to a person skilled in the art, especially because there are numerous examples of comparable process descriptions in the prior art which the prior art recognizes being adequate and understandable to a person skilled in the art. The specific value(s) of the procedure input data variable determine the chosen output port or route for the packet according to predetermined rules, so there is no ambiguity in any part of the processes described in my claims.

**Conclusion:**

**The claims have been amended to clearly and properly describe a patentable invention in simplest basic terms in a prior art recognized way, so they fulfill all conditions of patentability.**

.....